IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Christopher Lapkowski	Examiner: Tuan Vu	
Serial No.:	10/763,512	Group Art Unit: 2193	
Filed:	January 23, 2004	Docket: CA920030044US1	
Title:	PAIRING OF SPILLS FOR PARALLEL REGISTERS		

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Commissioner for Patents
P.O.Box 1450
Alexandria, VA 22313-1450

The Final Office Action mailed 8/14/2007 rejected claims 1-23 under 35 USC 103(a) as being unpatentable over Kolson[1] in view of Kahle.[2] However, Applicant submits that Kolson and/or Kahle do not teach or suggest each and every element of the claims.

Discussion of Claim 1

Claim 1 recites, "determining that register spill instructions in spill code generated by a register allocator can be associated with each other." The Office Action asserts that this claim feature is taught by Kolson's "register allocation unit analyzing what accesses can be made concurrent."[3] The Office Action attempts to support its assertion with three passages in Kolson.[4] The first two passages state:

> 1) Register allocation begins with the analysis of variable accesses in execution to derive the variable access stream... To denote concurrent accesses, parentheses bracket those reads or writes performed in parallel.[5]

> 2) Some variables will inevitably require memory accesses when updated or necessary for computation.[6]

For the third passage, the Office Action refers to Kolson's Section 5.1 and "register classes" without any explanation. Applicant submits the cited passages do not support the rejection. Although the first two passages may teach analyzing variable accesses and register allocation, none of the passages mention or elude to **spill instructions.** Moreover, the cited passages do not

1 David J. Kolson, et al. "A Method for Register Allocation to Loops in Multiple Register File Architectures" Proceedings of IPPP's 96, 1996, pp.28-33.
2 Kahle et al. US 5,867,648
3 FOA mailed 8/14/2007 Page 3, Number 4, "As per claim 1" section.
4 See Office Action at Page 3.
5 Kolson at page 28, right column (numbering added).

teach or suggest claim 1's "determining that register spill instructions in spill code generated by a register allocator can be associated with each other."

Claim 1 also recites, "based on the determining, rewriting said register spill instructions as a parallel register spill instruction." The Office Action asserts that Kolson's Figure 4 and passage in Section 5.2 teaches part of the above-cited claim feature (i.e., claim 1's "based on the determining, rewriting said register spill instructions"). However, Applicant submits that Figure 4 and its description in Section 5.2 illustrate and describe a function for mapping variables to registers. Although the description teaches that a variable might need to be spilled (i.e. initially generating spill code), Applicant can find nothing in the passage and figure that teaches or suggests **rewriting** register spill instructions, and particularly, rewriting register spill instructions based on analyzing them to determine whether they are associated with one another. As for the rest of the above-cited claim feature, the Office Action admits that Kolson does not disclose "rewriting said spill instructions as a parallel register spill instruction." The Office Action instead asserts that Kolson "discloses a [sic] architecture having multiple register implementation" and that "[a]nother alternative of using a multiple register operation as suggested in parallel read/write in Kolson's register extension approach... is further disclosed by Kahle's architecture. Kahle teaches using multiple register and multi-register architecture for load/store operations. However, the Office Action does not identify any passages in Kolson or Kahle that teach or suggest **rewriting** multiple spill instructions as a single parallel register spill instruction.

Claim 1 also recites, "based on said rewritten parallel register spill instruction, configuring storage of associated register spills in memory in such a manner that said register spills can be loaded back into said registers **in parallel**." (Emphasis added.) The Office Action asserts that Kolson teaches this in Figure 4's statement: "must be loaded from memory" (referring to variable values). However, Applicant submits that just because Kolson teaches that variables can be "loaded back," Kolson does not teach or suggest configuring storage of associated register spills in memory, so the load back can occur in parallel, as recited in claim 1.

For at least the reasons noted above, Applicant submits that the combination of Kolson and Kahle does not teach or suggest each and every element of Claim 1.

---

6 Kolson at page 29, section 3, right column (numbering added).

Discussion of Claims 9, 16, and 20

Claims 9, 16, and 20 each include features similar to those noted in the discussion of claim 1. For at least those reasons, Applicant submits that the combination of Kolson and Kahle does not teach or suggest each and every element of claims 9, 16, or 20.

Discussion of Claims 2-8, 10-15, 17-19, and 21-23

Claims 2-8, 10-15, 17-19, and 21-23 each depend, directly or indirectly, on one of claims 1, 9, 16, or 20. For at least the reasons noted above, Applicant submits that the combination of Kolson and Kahle does not teach or suggest each and every element of claims 2-8, 10-15, 17-19, or 21-23.

Claim Objections

Claims 1, 9, 16, and 20 are objected to because "the phrase recited 'manner that said register spills can be **loaded back** into said registers' does not appear in proper syntax because it does not establish any reasonable teaching that a prior register loading (to some registers) have been accomplished for a loading back to 'said registers' to make some sense... the underlying semantic of a *load back* without antecedent basis cannot be accepted." Applicant submits that a "register spill" occupied a register before it was spilled. Therefore, the register spill can be "loaded back" into the register. Applicant submits that the claim language does not lack antecedent basis and is clear to those of ordinary skill in the art.

Applicant requests that the rejections and objections be withdrawn and the claims allowed.

Respectfully submitted,

CHRISTOPHER LAPKOWSKI

By his Representatives,

DELIZIO GILLIAM, PLLC
15201 Mason Road
Suite 1000-312
Cypress, TX 77433
281-758-0025

Date 12/14/2007 By /Andrew DeLizio, Reg # 52806/
Andrew DeLizio
Reg. No. 52,806

Date of Deposit: December 14, 2007
This paper or fee is being filed on the date indicated above using the USPTO's electronic filing system EFS-Web, and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.